| Feedback Examples |
|---|
| : "Excellent work Jeremiah. Always nice to see you take initiative and go beyond the requirements. ", |
| : "Great work!", |
| : "Great work! One suggestion is to try to devour the burgers individually rather than all at once.", |
| : "I saw that you changed your code to sequalize however you app doesn't present a match. It just shows me a modal that with no name for the match or picture. \n\n", |
| : "Awesome work! Great job with sequalize  and the app in general. I think if we can get some of bugs worked out RCB may be able to put this into production.", |
| : "Good work, but I could not submit a new Song. It keeps giving me an application error, so I can't tell if your sequalize queries/connection are working properly. Looking at your sequlize queries though it seems your have set them up properly. ", |
| : "Your on the right track in terms of using sequelize. However, you are missing a `post` route for the `/` path. This is where you'll want to manage what happens when the `eat me` button is clicked.\n\nThis is also the reason you are not seeing an update to the page when you add a new burger, but it shows up on a reload.", |
| : "Hey Walter, it seems like you have a bit of work to do on this. If you want to sit down and discuss your HW before or after class let one of us know.", |
| : "You are on the right track for the most part. \n\nRegarding the `Devour It` button, your console log shows the same burger (Hamburger) no matter which `Devour it` button you click. This is because the`id` that you are referencing in your `.devour on click` is the same for all of them. ([See Line 76](https://github.com/ltdelia/SequelizeBurgerTime/blob/master/views/index.handlebars#L76)). \n\nHere are the steps I took to get your started in terms of getting the correct burger name on click.\n\n* Add an id on the devour button that is dynamically generated. So you can grab it later in the on click event. This is at [Line 27](https://github.com/ltdelia/SequelizeBurgerTime/blob/master/views/index.handlebars#L27)\n\n```\n<td><button id={{this.burger_name}} type=\"submit\" class=\"devour\">Devour it!</button></td>\n```\n\n\n* Update your devour on click event to capture the new `id` on [Line 74-76](https://github.com/ltdelia/SequelizeBurgerTime/blob/master/views/index.handlebars#L74-L76)\n\n```\n$(\".devour\").on('click', function(event){\n\n\tvar burger = {burger_name: event.target.id};\n```\n\n* Next you'll need to update the database by setting devoured to true.", |
| : "Hey Stephen, can you update your link. It points to a local dir not your github or heroku", |
| : "Hey Neena, the link to both your heroku and github repo are not working. Can you update them as soon a possible?", |
| : "Hey John, we should set some time aside either before after class and discuss this HW so you are comfortable with sequelize.", |
| : "Excellent work!", |
| : "Aaron,\n\nThis is your burger app from week 14 dude! Lemme see your week 15 hw!\n\nThanks!", |

| |
|---|
| : "Aisha you seem to have your sequelize queries properly constructed. However, your heroku link does not work, and when I ran your application locally there were errors with your database. It would be a good idea to set aside some time before or after class to discuss how to get this working 100%.", |
| : "Good work with your sequelize queries. You were on the right track and had some minor issues which prevented your app from working properly. Always check console/terminal to see what errors your are getting. This will make debugging much easier.\n\nThere were 3 things that needed to be fixed:\n\n 1. The  sequalize settings in your [connection.js](https://github.com/amanakiwala/starwars_homework/blob/master/app/config/connection.js#L32-L42) file needed be adjusted. Sequelize automatically tries to add timestamps to your tables & queries. To prevent this you need to tell not to do this by using `timestamps: false`.\n\n```\n// Creates mySQL connection using Sequelize\nvar sequelize = new Sequelize(selectedSource.database, selectedSource.user, selectedSource.password, {\n define: {\n        timestamps: false\n    },\n\n    host: selectedSource.host,\n  dialect: 'mysql',\n\n pool: {\n    max: 5,\n    min: 0,\n    idle: 10000\n  },\n\n});\n```\n\n 2. In your model [character.js](https://github.com/amanakiwala/starwars_homework/blob/master/app/model/character.js#L23) the table definition had an error. In your MySQL Database you have a column named `forcePoints` with an `s`. However in your model definition you have it as `forcePoint` without the `s` at the end. \n\nThis is how that line should look:\n\n```\n    forcePoints: { \n```\n\n 3. In the [connection.js](https://github.com/amanakiwala/starwars_homework/blob/master/app/config/connection.js#L29) file the source database needs to be updated to jawsDB instead of localhost.\n\nAfter changing these things your app works correctly.", |
| : "Jerome, I think you had issues moving from DB to another. You seem to be referencing columns in your sequelize queries that probably refer to your local MySQL DB. However your Amazon DB has different columns and names.", |
| : "Hey Jasmine, can you update the `seeds.sql` file so I have some data to test on. Thanks!", |
| : "I have graded your assignment based on what I can gather from your code on github. Your sequelize queries seem to be on point, however I couldn't test them. ", |
| : "Hey Adam,  from what I can tell You were still using the methods you created with your orm.js. Since you are using Sequelize now, you should use the methods that are given to you by that orm. So for example when using selectAllBurger referencing the orm object you created, instead try burger.findAll(which all sequelize objects will have when using sequelize as your orm). Also when you defined a new Sequelize object called Burgers make sure to be consistent with the naming and capitalization when using that model in your burger_controller.js file. If you have any questions or need help reach out to me!", |
| : "Works great! Awesome job!\n\nOnly thing I noticed were the overflow issues on your `main.handlebars`.\nTo fix this change line 14 to `<div class=\"header text-center container-fluid\">` and change line 44 to `<footer class=\"container\">`. You just need some container classes around your rows.\n\nKeep up the great work!", |
| : "Hey Raghav -- I can't open the folder in your repo. Let me know when you fix this if you want feedback!", |
| : "Hey James!\n\nI've been sending the same comment to a lot of students so I've made a gist for you to use as a cheatsheet:\nhttps://gist.github.com/dannyvassallo/89a89670693557974bf02b209507077f\n\nBesides that -- Looks like its working great! Keep up the good work!", |

| |
|---|
| : "Hey man, For your /add route, when you defined the Schema object make sure to reference the same table within your schema.sql file. Also when you are getting input from user via req.body make sure in the html youre using name=\"\" to get input from user and that is what youll use to save an instance of Characters into your database.\n\n Also try to be consistent with your naming, just use whatever you defined as parameters in your sequelize object to reference when you are saving to database using user inputs. Alot of the times you were trying to save a parameter role into the model where you defined it as type!\n\nFor your view.html to search for specific pokemon, in your if statement data == false should be instead data === undefined and the div will show up when you cant find a specific pokemon in your database.\n\nAlso are you having trouble with handlebars or did you just forget to add it into your project?\n\n If you have any questions please ask! And great job on this you were able to understand the concept but just a couple  blunders here and there caused some bugs. ", |
| : "Nice job dude!", |
| : "Hey man, How you have your routes setup atm you are still using the orm methods to update or find your burger data. You instead want to use the Sequelize terminology and use their methods, for example your Burger.SelectAllBurgers can simply change to Burger.findAll.  If you have any other questions feel free to ask me.", |
| : "Hey man, Looks good everything works for the first part when you try to make a reservation. Did you just not have time to implement the waitlist function? Just want to make sure you understand concept, if you have any questions feel free to ask", |
| : "Hy did you not get this Hw or did you not get around to doing it?", |
| : "Hey Eric, When youre setting up your connection in connection.js make sure to include the right package. You were using mysql instead of sequelize!  looking over your routes you were still using the ORM, did you jsut not have a chance to rewrite into Sequelize? if you have any questions hit me up.", |
| : "Nice, dont forget to deploy! Nice job on the clear table functionality.", |
| : "Hey shawn -- do you have schema or seed file I can migrate to populate my database? If not, I can create a matching one. I just figured this could expedite things.", |

: "Hey Kev!\n\nThis doesn't seem to be working, but you are on the right track.\n\nTo get you started, I removed the `orm.js` file, and created a models directory.\nWithin the models directory I setup a `Products.js` model and a `Departments.js` model.\n\nThe `Products` looks like this:\n\n```\nvar Sequelize = require(\"sequelize\");\nvar connection = require(\"../connection.js\");\n\n// initiate the products table\nvar Products = connection.define(\"products\", {\n  itemId: {\n    type: Sequelize.INTEGER,\n    field: \"Item_ID\",\n    allowNull: false,\n    autoIncrement: true,\n    primaryKey: true\n  },\n  productName: {\n    type: Sequelize.STRING,\n    field: \"Product_Name\",\n    allowNull: false\n  },\n  deptName: {\n    type: Sequelize.STRING,\n    field: \"Department_Name\"\n  },\n  price: {\n    type: Sequelize.FLOAT,\n    field: \"Price\"\n  },\n  stockQuantity: {\n    type: Sequelize.INTEGER,\n    field: \"Stocky_Quantity\"\n  },\n  createdAt: {\n    type: Sequelize.DATE,\n    allowNull: true,\n    defaultValue: Sequelize.NOW\n  },\n  updatedAt: {\n    type: Sequelize.DATE,\n    allowNull: true,\n    defaultValue: Sequelize.NOW\n  }\n},\n{\n  freezeTableName: true // model tableName will be the same as the model name\n});\n\nProducts.sync({force:false}).then(function (err) {\n  if(err){\n    // console.error(\"ERROR on table creation\");\n    console.error(\"ERROR\" + err);\n  }else{\n    console.log(\"Table Created Successfully\");\n  }\n});\n\nmodule.exports = Products;\n```\n\nNow, I can reference the models by name in `BamazonCustomer.js` using the following:\n```\nvar Products = require(\"./models/Product.js\");\nvar Departments = require(\"./models/Department.js\");\n```\n\nAnd take advantage of sequelize's helpers like `Products.findAll()` to return all of the Products from the database.\n\nThis alleviates the need for the custom queries you have setup from your original ORM.\n\nI hope this helped!",

: "Hey Jeremy -- you pretty much knocked it out of the park with this!\n\nin reference to your comment here:\n```\n// I am not sure what is meant in the homework instructions\n// by 'burger-specific input', and why this module\n// needs to exist at all.\n```\n\nRather than creating multiple models in an ORM, you create individual js files for each model. In this case it didn't necessarily apply since you refer to a solitary model. \n\nYou have:\n\n```\nvar burgers = DB.define('burgers', {\n  id:       { type: Sequelize.INTEGER, primaryKey: true},\n  burger_name: Sequelize.TEXT,\n  devoured: Sequelize.BOOLEAN,\n  the_date: Sequelize.DATE,\n},{timestamps: false});\n\nDB.sync();\n\nmodule.exports = burgers;\n```\n\nin your `connection.js`. You can move this to a model in `models/burger.js` and when you require it, you can reference the model name explicitly. After requiring you should be able to make queries like `Burgers.findAll();` to return all burgers. Check the sequelize documentation on \"Models\" for more information.\n\nGreat work though!",